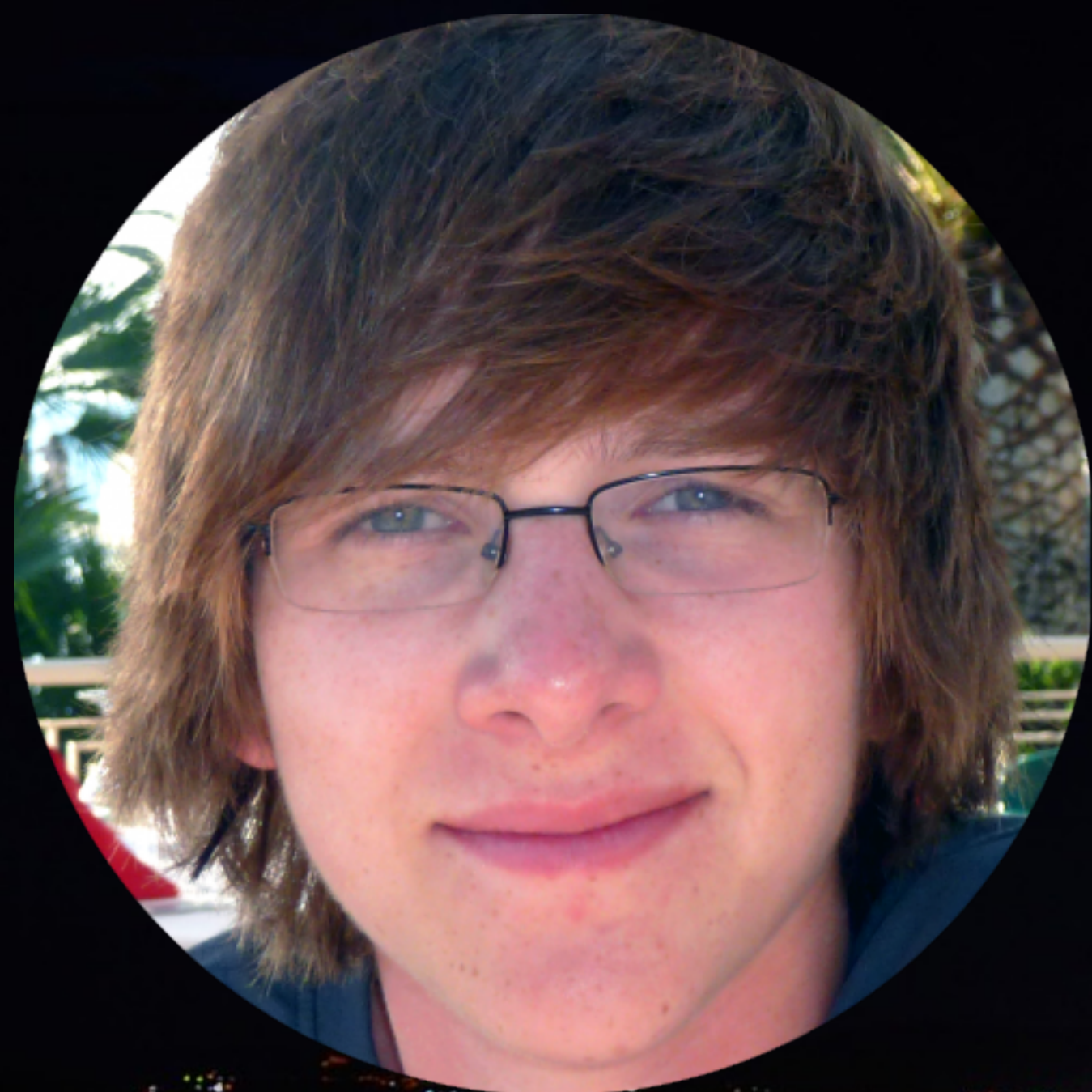


# JavaScript

for PHP developers

JS





Daniel Laxar  
@dlaxar





# JavaScript

for PHP developers

JS

Let's be honest...

JS

JavaScript is weird.

JS

# What this talk is about

Showing ways to  
deal with JS

JS

# What this talk isn't about

Make you like JS

Basics of JS

Make you a JS Pro

JS

# Today's Agenda

Facts

Code Structures

Gottchas

node.js

JS



# JavaScript

developed in 10 days

Client Side scripting language

first seen in Netscape

runs in a VM

works without jQuery

incompatible as hell

JS

# PHP

obj->prop

class::prop

# JS

obj.prop

thing.prop

# JS

# Code Structures

IFFY

Namespaces

Objects

Clojures

JS

# IFFY

Immediately **I**nvoked **F**unction **E**xpression (**IFFY** - wait... what?)

# JS



# Why do that?

Avoid polluting global space

Enables private variables/functions

JS

# IFFY

```
(function() {
```

```
doStuffAndSoOn();
```

```
})();
```

Function Expression

= Anonymous Function Definition

Brackets

Function Invocation

= Invoke (Brackets = Function)

JS

# IFFY in the wild #1

```
(function(window, document, undefined) {'use strict';  
  
    /* Angular.js defined here */  
  
})(window, document);
```



# IFFY#1 - the point in doing so

```
var undefined = 'defined'; /* evil code */
```

```
var o = {};
```

```
if(o.prop === undefined) { /* dead code */ }
```

```
(function(undefined) {
```

```
    var noPollution = 'solar-energy'; /* not leaked globally */
```

```
    if(o.prop === undefined) { /* works */ }
```

```
})();
```

JS



# IFFY in the wild #2

```
(function( global, factory ) {  
  
    factory(/* ... */);  
  
} )(  
    typeof window !== "undefined" ? window : this,  
    function( window, noGlobal ) { /* jq defined here */ }  
);
```

JS

# Namespaces

`\Human\Brain\Memory` vs `\Computer\Mainboard\Memory`

`Human.Brain.Memory` vs `Computer.Mainboard.Memory`

JS

# Namespaces

```
// global (i.e. no namespace)  
window.Memory = something;
```

```
// sane (i.e. with namespace)  
window.Human = window.Human || {};  
window.Human.Brain = window.Human.Brain || {};  
window.Human.Brain.Memory = something;
```

```
// or  
window.Human.Brain.Memory = window.Human.Brain.Memory || {};
```



# Namespace in the wild #1

```
goog.require('goog.dom');
```

```
function sayHi() {  
    var newHeader = goog.dom.createDom('h1',  
        {'style': 'background-color:#EEE'},  
        'Hello world!');
```

```
    goog.dom.appendChild(document.body, newHeader);
```

```
}
```

JS



# Namespace in the wild #2

```
d3.map = function(object) {  
    var map = new d3_Map();  
  
    /* ... bla foo */  
};
```

JS

# Objects

i.e. Classes

JS

# Objects

“Classes” are Functions

The Functions are the Constructors

Every Object has a Prototype (i.e. is an Instance of a Prototype)

A Prototype is the recursive list of methods an object has

JS

# Objects

```
var Car = function() { /*this is the constructor*/ };  
  
Car.prototype.drive = function() { /* a 'method' */ };  
  
var Ferrari = function() { /* this is another constructor */ };  
  
Ferrari.prototype = new Car(); /* Ferrari inherits Car */
```





# Closures

i.e. doing some magic with variable scopes

JS

```
(function Outer(g) {  
    var string = 'Hello, World';  
    function Inner() {}  
    Inner.prototype.getPrivate = function() { return string; };  
    g.Inner = Inner;  
})(window);
```

```
(new Inner()).getPrivate();
```

JS

# Gottchas

JavaScript doesn't like you. Face it.

JS

```
var a = 2;  
function test() {  
    console.log(a);  
    var a = 5;  
    console.log(a);  
}  
test();
```

// Output?

JS

```
var a = 2;  
function test() {  
    var a = undefined;  
    console.log(a);  
    a = 5;  
    console.log(a);  
}  
test();
```

// Output?

JS

# Hoisting

```
var a = 2;  
function test() {  
    var a = undefined;  
    console.log(a);  
    a = 5;  
    console.log(a);  
}  
test();
```

// Output: undefined 5



```
var a = ["a", "b", "c"];

for(var i = 0; i < a.length; i++) {
    setTimeout(function() { console.log(a[i]) }, 10);
}

// Output?
```

JS



```
var a = ["a", "b", "c"];
```

```
var i = 0;
```

```
for(; i < a.length; i++) {}
```

```
console.log(a[i]); console.log(a[i]); console.log(a[i]);
```

```
// Output?
```

JS

# Context

```
var a = ["a", "b", "c"];
```

```
var i = 0;
```

```
for(; i < a.length; i++) {}
```

```
console.log(a[i]); console.log(a[i]); console.log(a[i]);
```

```
// Output: c c c
```



# node.js

server side JS

going to kill PHP in 2012

Google's V8

not the first server side JS

JS

node.js

JS

# node.js

Slideshows suck.

# JS

JS

# JavaScript

for PHP developers

JS



# JavaScript

JS

# JavaScript

return **this**;

JS